A Web Content Suggestion System for Distance Learning

Che-Yu Yang[1]*, Hun-Hui Hsu[2] and Jason C. Hung[3]

[1]*Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C.*
[2]*Department of French, Tamkang University, Tamsui, Taiwan 251, R.O.C.*
[3]*Department of Information Management, Northern Taiwan Institute of Science and Technology, Taipei, Taiwan 112, R.O.C.*

Abstract

Nowadays the information technology is developing quickly, the information technology changes with each passing day. The accessibility of the World Wide Web and the ease of use of the tools to browse the resources on the web have made this technology extremely popular and the method of choice for distance learning.

There are many studies have been done regarding distance learning, many studies focus more on system instruction and in curriculum organization. These do not help the E-learner to study according to his or her interest. Nevertheless, the goal of this research is somewhat different from other works.

This research tries to make the extension of the lecture teaching. The instructor prepares the course content related information as supplementary reading or knowledge, and composes them into the format of web pages, and the online students can browse the web content. Then, according to students' navigation activities and the properties of the web content, our system can suggest the web content might interest them but has not been read to them.

Our Web Content Suggestion System for Distance Learning makes suggestions to students in two ways: (1) collaborative filtering — matches one's tastes or needs to those of the other students who share his likes and dislikes and suggests web content that they have read and liked but he hasn't read; (2) content-based filtering — suggests web pages similar to those one prefer based on a comparison of web content.

The web content suggestion can enlarge student's learning space and widespread their interests that lecture teaching hardly offers. Through this system, we can further dig out or induce students' interests, thus widen their knowledge scope.

***Key Words*:** Distance Learning, Collaborative Filtering, Content Based Filtering, Web Mining

1. Introduction

The technology of web mining can be helpful in constructing mass customization on web content. In addition, it can be useful in identifying virtual knowledge structure in a web-based distance education. Web mining can be broadly defined as the discovery and analysis of useful information from the World Wide Web.

The term "Web mining" has been used in two distinct ways [1]. The first, called Web content mining, is the process of information discovery from sources across the World Wide Web. The second, called Web usage mining,

*Corresponding author. E-mail: yang@schummi.com

is the process of mining for user browsing and access patterns. In this research, we combined content-based filtering and collaborative filtering to design our web content suggestion system for distance learning. Content-based suggestion delivers web content (web pages) to the students who have navigation records, on the other hand, collaborative suggestion clusters students into groups, students in the same group are of similar behavior or interests.

The main idea of this research is to collaborate with the lesson topics and web page resource to extend students' learning space. Instructor publishes supplementary reading or course relative extracurricular content to the learning system website. Students can browse and read the web pages according to their interests. Through the utilization of information technology, and the content of the web page resource, we can not only provide what may be insufficient in the classroom teaching, but also through the automated web content suggestion procedure, we can extend students' interests and enrich their knowledge.

For each student participating in our online learning system, there is a personal record for him/her, the personal record consists of *personal profile*, such as name, age, degree of foreign language, interests..., etc. And also, the personal records store the web page *navigation behavior* of students. The information in the record will then be utilized by our web content suggestion system to analyze students' behavior, and make suitable suggestion on web content to the students.

Basically, we divide the web pages of supplementary reading or course relative extracurricular information in our learning website into several topics. For example, as a course on French language learning, we may have topics of French food, vogue, travel, geography, education, entertainment, culture, economics, politics…, etc. Students can browse these topics according to their own interests or needs, and these browsing activities will be captured and stored as the navigation record in the personal records. Then, we can partition students into groups according to their navigation records. For example, a group of students interested in French food, another group of students interested in French politics.., etc. Nevertheless, students in the same group are not necessary to have browsed exact the same web pages, therefore, we can actively suggest some web pages that they may be interested in but have not read yet.

For those students who are new to attend the course and have limited navigation record, we may first ask them to poll for their interests and then temporarily assign them into group according to their personal profile. This is also based on the observation that people with similar background may have similar interests. Later, these will be adjusted and updated referring to their navigation records. Besides, for those web pages very popular in a topic, the system will also suggest them to students in other groups. So that we can further dig out or induce students' interests, thus widen their knowledge scope.

In the next section, we introduce the related searches. The web content suggestion system architecture is demonstrated in section 3. The Webpage-Keyword-Student 3D Information Model is proposed in section 4. The Student Clustering and Navigation Pattern Discovery algorithm is delineated in Section 5. The events that trigger the suggestion are discussed in Section 6. Finally, a conclusion is drawn.

2. Related Work

In a typical information filtering system, each user is associated with a *profile* which contains his interests, background, needs…, etc. By comparing the data with these profiles, the users interested in the data are recognized and informed. This idea as been applied to various information systems on the Internet, such as SIFT [2] and SIFTER [3]. Instead of searching data for users like the information retrieval systems, the information filtering systems find the matched profiles for given data. For that reason the filtering systems save the costs on data indexing and have better performance when the volume of data grows while comparing to the retrieval systems. Furthermore, the filtering systems also intend to provide users a way to share information with others. Many of the researches focused on the issue of how to select the matched profiles for given data. There are two kind of major approaches: "content-based filtering" and "collaborative filtering".

2.1 Content-Based Filtering

This approach recognizes and provides the relevant data for the users based on the similarity between data and profiles. The text retrieval techniques [4] can be applied to profile indexing. Since the content-based filter-

ing approach is powered by contents, its effectiveness depends on the well description of user's information needs in the profile. If the users fail to provide correct and precise descriptions, the filtering task may lead to undesired results. Furthermore, the content-based filtering approach is not able to provide unexpected but interested discoveries for the users.

In the fields of Web application, the web content is the real data and the web page was designed to convey to the users [5]. It consists of several types of data such as unstructured text, graphics, sound, video and semi-structured hypertext. Content mining can be referred to as the application of data mining algorithms to the content of the web [5]. A conceptual schema can be created [6] that can describe the semantics of a large volume of unstructured web data to manage them. [7] discussed various categories of the web content mining such as text mining which is mining of unstructured texts and multimedia data mining which is mining of multiple types of data such as unstructured and image data.

2.2 Collaborative Filtering

This approach first recognizes relevant users who have similar profiles, then provides data they like to each others. This approach measures the similarity between profiles rather than between data and profiles. The major issue in this approach is how to cluster the user profiles for effective filtering. There are some methods [8,9] that can incrementally cluster multi-dimensional data points. Since the collaborative filtering is powered by user clustering, its effectiveness depends on whether the clustering of profiles correlates the users well. If meaningful user clusters are not derived, then the filtering process may bring undesired results. Furthermore, this approach may provide unexpected findings for uses from its essence of information sharing.

In E-Commerce, the analysis of web visitors' behavior can give important clues about current market trends and help merchants to predict the future trends of potential customers. Analysis of long visit-paths of users may facilitate the need of restructuring of the website to help visitors reach desired information quickly. Also, the mined knowledge can be used to offer preferred web content to visitors.

A lot of research work has been conducted on web personalization. Adaptive Sites use information about user access patterns to improve their organization and presentation for different types of users [10]. A technique for capturing common visitor profiles using association rule discovery and usage-based clustering of URLs is proposed by [11]. A technique for web personalization proposed in [12] is based on association rule discovery from usage data. An algorithm to reorganize a web site using page access frequency and classification of pages is proposed by [23]. A tool is developed and described in [13] for customizing the website dynamically.

[14] performed association rule mining to discover interesting behavioral patterns of mobile device users. Finding page locations that are different than where visitors expected them to be during their visit also helps to restructure website organization [15].

Many researched utilized the filtering approaches to facilitate personalizing the system interactions with users. Example applications range from keystroke prediction [16] and TV listing [17], to web navigation [18] and web search [19]. Some researches allow users to comment on Netnews web pages and then get the ones recommended by other users. In theses systems users have to specify their profile explicitly to get the recommendations. However it is not so convenient for the users who often change their interests to update their profiles repeatedly. In this circumstance, a way to derive the user profiles implicitly will be useful. The machine learning and data mining [20] techniques are common used. On the Web, user requests are usually logged as the navigation history. The navigation histories are good source to indicate what the users want [21,22], and to derive the user profiles.

3. Content Suggestion System Architecture for E-Learning

The content suggestion system we proposed consists of 6 components — Student Assistant Agents, Student Identification Component, Student Behavior Generation Component, Suggestion Generation Component, Suggestion Delivery Component and Data Warehouse. Figure 1 is the system architecture, and the function of each component is described as follows:

3.1 Student Assistant Agent

At the client side, we deployed the *Student Assistant Agent (SAA)* to capture student's behavior. The SAA is a

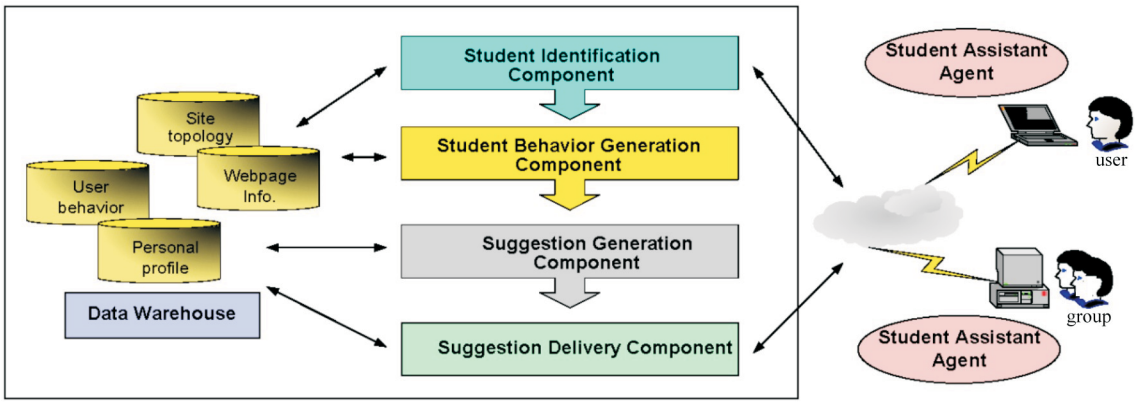

Figure 1. The content suggestion system architecture.

mobile agent that helps students to find their favorite web pages and content. On the other hand, if students have no idea what to browse, they can indicate their interests or needs to the agents. After the computation at the server side, the agents will suggest suitable content to them. Besides acting as a user interface, the SAA also captures student's navigation behavior for further behavior analysis at the server side.

A web page navigation history record, the *WPN record*, which stores navigation activity information, will be established by SAA for each student. The WPN record is a set defined as:

$WPN_{i,s} = \{p_1, p_2, \ldots\ldots p_m\}$, where $p_1, p_2, \ldots p_m$ are the web pages that user i navigated in the session s.

As we will extract keywords for each web page in the website, another record — the *keyword focusing record (KFR)* — will be established by SAA too. The KFR record stores the information about what keywords and how much (the degree) a student focuses on. The KFR record is a set defined as:

$KFR_{i,s} = \{F_1, F_2, \ldots\ldots F_n\}$, where F_n is defined as a record $\{k_{id}, stime, etime\}$, k_{id} is an identified keyword in the web pages. Where *stime* and *etime* are the times that user starts to focus on and ends focusing on the keyword (the times user start and finish the navigation on a web page that contains the keyword).

We will use the WPN and the KFR records to build a web content and student behavior information model (the WKS model) later.

3.2 Student Identification Component

In our system, students must first register as a member before they can enter the site and login into the system. This component verifies student's privilege and identification. Besides, this component communicates with the SAA and monitors whether the agents are alive or not. In this component, we must insure that the correct profile belongs to the specific student. This is essential for 1-to-1 personalized suggestion.

3.3 Student Behavior Generation Component

After students enter the web site, they can navigate the web pages. In this component, we record these browsing behaviors, filter these data into meaningful information, and then store them in the Data Warehouse. This information would be very helpful to find the students' interests.

3.4 Suggestion Generation Component

This component analyzes students' behavior information generated by the Student Behavior Generation Component, and clusters students into several groups. Students in the same group have similar behavior or interests. The system then uses the correlation information between groups to make suitable content suggestions to the students. We will explain our student behavior analysis algorithm in the later section.

3.5 Suggestion Delivery Component

After the suitable contents are selected by the Suggestion Generation Component, the next task is to deliver the suggested content to students. The Suggestion Delivery Component delivers suitable suggestion to the proper students. Besides, this component can also suggests to instructors for better web content construction.

3.6 Data Warehouse

In our system, there are huge amount of information access taking place, so it is critical to establish a good database management mechanism. The Data Warehouse stores students' behavior records, personal profiles, the site topology, and the webpage information…, etc.

4. The Webpage-Keyword-Student 3D Information Model

According to the students' web page navigation records, we further build a three-dimension information model — the *WKS Model*. Each dimension consists of meaningful information about student behavior or web page content which can be further processed to extract useful information for generating content suggestions. Figure 2 is the logical view of the WKS model.

We explain each "view" of the model as below (A "view" is a matrix form by the combination of any of two axes).

4.1 Student-Webpage Matrix

Table 1 is the logical view of the Student-Webpage matrix. If a student *i* navigated a web page *j*, then the S*i*P*j* will be filled with '1', otherwise, it would be filled with '0'.

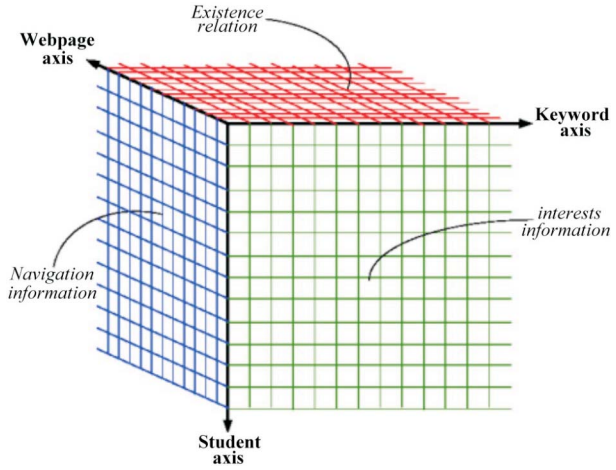

Figure 2. The webpage-keyword-student information model.

Table 1. Student-webpage matrix

		Web page Identification					
Student Identification		P_1	P_2	P_3	...	P_n	
	S_1						SCI_1
	S_2						SCI_2
	⋮				S_iP_j		
	S_m					S_mP_n	SCI_m
		PCI_1	PCI_2	PCI_3	...	PCI_n	

For students, we can use the *SCI* to find which students have widespread interests; for web pages, we can use the *PCI* to find which web pages are potentially popular web pages. The *SCI* and *PCI* are calculated as equation 1 and equation 2:

■ Student Caution Index (SCI)

$$SCIi = \frac{\sum_{j=1}^{n} SiPj}{n} \tag{1}$$

where S_iP_j is the matrix's content, and n is the total number of the web pages in the web site.

■ Webpage Caution Index (PCI)

$$PCIj = \frac{\sum_{i=1}^{m} SiPj}{m} \tag{2}$$

where S_iP_j is the matrix's content and m is the total number of the students.

4.2 Keyword-Webpage Matrix

The logical view of the Keyword-Webpage matrix is similar to the logical view of the Student-Webpage matrix.

Each element in the Keyword-Webpage matrix is a compound value consists of two properties - $K_iP_j = \{V_{ij}, S_{ij}\}$.

V_{ij} is a Boolean value that indicates some web pages have some keywords that represent the content of the web pages. If web page *i* has keyword *j*, then the K_iP_j will be filled with '1', otherwise, it would be filled with '0'. This matrix is automatically established by information retrieval technique. The value of V_{ij} is calculated as the following procedure.

We use words as the basic units to represent text in the web pages. A "word" is defined as a contiguous string of characters delimited by spaces. Specifically, each web page is preprocessed in the following steps:

(1) Punctuation marks are separated from words.
(2) Numbers and punctuation marks are removed.
(3) All words are converted to lower case.
(4) Words like prepositions, conjunctions, auxiliary

verbs, etc., are removed. The remaining words after preprocessing are potential candidates for use as keywords.

(5) Refine the keywords with Log-Entropy weighting scheme.

Log-Entropy is based on information theoretic ideas and is the most sophisticated weighting scheme. Log-Entropy weighting scheme assigns minimum weight to terms that are equally distributed over documents, and maximum weight to the terms that are concentrated in a few documents. Entropy takes into account the distribution of terms over documents. We eliminate the lower half weighted keywords, and left the upper half. The equation 3 is log-entropy formula.

$$Entropy(T) = 1 - \sum_{j=1}^{n} P_j \cdot \log_2 Pj \quad (3)$$

where T is a term in the collection, P_j is "the frequency of term T in document j" divided by "the total number of times term T occurs in the whole collection."

The other value of the compound K_iP_j:S_{ij}, is calculated as:

$$S_{ij} = \frac{total\_navigation\_time\_of\_page\_j}{total\_navigation\_time\_of\_keyword\_i} \quad (4)$$

4.3 Student-Keyword Matrix

This matrix indicates that someone is interested in a web page and focuses on the web page's keywords. In the previous section, we mentioned that the SAA builds the WPN and KFR records for each student; now, we further extract these records into the *navigation vector*. Each student has a navigation vector respect to a web page. According to the navigation vector, we compare students with each others to find their correlations, and then cluster the students into heterogeneous groups. A navigation vector of a web page for the student i looks like: $NVi = (k_1, k_2, k_3..., k_n)$, where the web page has n keywords. Each element in the navigation vector indicates the interest of the student to a particular keyword of the web page. The value of a navigation factor Nf_k is a Boolean. If the page has keyword k, then the value of Nf_k is "1", otherwise the value is "0". The element k in navigation vectir is calculated by equation 5:

$$navigation\_vector\_element\_{\_k} = \frac{\sum Nf_k}{K} \times avg(Nfk) \quad (5)$$

where $avg(Nf_k) = \frac{time\_of(Nf_k)}{total\_time(P_j)}$, k is the length of the navigation sequence of web page j which student i navigated, $time\_of\ (Nf_k)$ is the total time the student navigated the keyword k, $total\_time\ (P_j)$ is the total time the student navigated the web page j.

And we use Final Navigation Vector to update the student's latest preference:

$$FNV_i = old(FNV_i) \times W_1 + (k_1, k_2, k_3, \ldots k_n) \times W_2 \quad (6)$$

where $W_1 + W2 = 1$, $(k_1, k_2, k_3, \ldots k_n)$ is the new FNV value of the user i. W_1, W_2 is a set of factors which is adjusted according to the student's response to the suggestion. If the student accepts our suggestion, we will enhance the factor W_2; if the student doesn't accept the suggestion, the factor W_1 will be enhanced. This approach is the basic evaluation of our system and the fundamental method to update the student profile.

For instance, student i navigated the web page p and the navigation sequence is shown in Figure 3. The navigation graph is an extended sub-graph of the site topology. In Figure 3, a student i, had a navigation sequence {a, b, c, d, c, e, d, e, f}, can be observed. This sequence is captured in a connection session.

According to the data in Keyword-Webpage Matrix (Table 2), which shows some web pages have some keywords and the navigation time, we can further construct this navigation sequence into Table 3.

Thus, there are nine navigation vectors:
(1, 1, 0, 1, 0, 0, 0), (1, 0, 0, 1, 1, 0, 1), (0, 1, 1, 0, 0, 0, 1), (1, 0, 0, 1, 0, 1, 1), (1, 0, 0, 1, 1, 0, 1), (0, 1, 1, 0, 0, 0, 1),

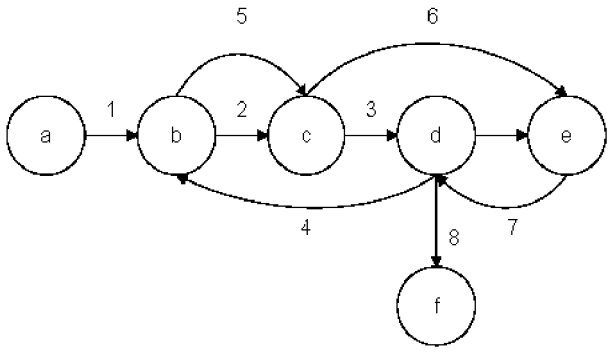

Figure 3. An example of student navigation sequence.

Table 2. Keyword-Webpage matrix

Keyword Identification	Web page Identification				
	P_1	P_2	P_3	...	P_n
K_1					
K_2					
⋮				K_iP_j	
K_q					K_qP_n

Table 3. Keyword navigation table derived from a web page navigation sequence

Sequence slice	Keyword ID	Navigation time
(0,a)	1,2,4	15
(1,b)	1,4,5,7	13
(2,c)	2,3,7	17
(3,d)	1,4,6,7	20
(4,b)	1,4,5,7	7
(5,c)	2,3,7	10
(6,e)	4	9
(7,d)	1,4,6,7	9
(8,f)	3,5	10

(0, 0, 0, 1, 0, 0, 0), (1, 0, 0, 1, 0, 1, 1), (0, 0, 1, 0, 1, 0, 0)

Then a navigation vector: (0.323, 0.127, 0.112, 0.442, 0.091, 0.059, 0.461), is calculated by equation 5 for student *i* in the session.

Table 4 is the logical view of the Student-Keyword matrix. Each value in the matrix is calculated by the above procedure. Each student's row is corresponding to a navigation vector that can be used for computing the similarity with other students.

5. Student Clustering and Navigation Pattern Discovery Algorithm

The clustering approach we adopt is based on the *k-means clustering* algorithm. We cluster students according to the information from the Student-Keyword Matrix of the WKS model, and the result is each web page has a group of students who are interested in it. During the clustering procedure, the experience value of the cluster number is initialized and then the students would be clustered into numbers of groups, where the students in the same group would have similar navigation behavior or interests. We construct *Algorithm 1* for clustering students according to their behavior on a specific web page.

Table 4. Student-Keyword matrix

Keyword Identification	Student Identification				
	S_1	S_2	S_3	...	S_n
K_1					
K_2					
⋮				K_iS_j	
K_q					K_qS_n

Algorithm 1: Generate Student Groups for Web Pages

Input: Student-Keyword Matrix, k is the cluster number, n is the number of students

Output: Student Groups

```
for i = 1 to k
    Gi ← i
    do
    {
        for i = 1 to n
        {
            most ← ∞
            for j =1 to k
            {
                temp = sqrt( sum_{z=1}^{m} |S_i K_z − S_j K_z|^2 )
                if temp < most then
                {
                    index = j
                    most ← temp
                }
            }
            Gindex ← Gindex ∪ i
        }
        for i = 1 to k
        {
            temp ← 0
            for j = 1 to Gi.length
                temp ← temp + G_{i,j}
            Gi ← temp/Gi.length
        }
    } while set G changes
output G
```

The Generation of Suggestion procedure suggests web pages to students based on the student groups. The clustering method is revised from the K-means clustering algorithm. The procedure generates several groups with respect to student's navigation mean vector and each

student belongs to one of the generated group. After generating the groups of students, the rating probability of each web page would be computed. If the rating probability of a page is larger than a threshold, then the page would be added to the top page list. If a student in the group does not read the ones in the top page list, the pages would be added to the suggestion list of the student.

On the other side, the navigation pattern information tends to keep in a more constant way. *A priori* algorithm is a method for finding frequent itemsets in traditional knowledge discovery field, and is often used to find the association rules in a large database. In the below we proposed an extension of the A priori algorithm for the discovery of student navigation pattern. We use *Algorithm 2* to find student's navigation pattern.

Algorithm 2: Discover Navigation Pattern

Input: Student-Webpage Matrix, minimum support count
Output: Navigation Patterns

```
for each Web page Pj
{
    count ← 0
    index ← j
    for each Si
        if SiPj = 1 then
            count ← count + 1
        SSindex ← j
        SSindex.count ← count
}
max ← index
do
{
    for i =1 to max
        max_temp ← max
        for j = i to max
            count < -0
            for each Sk
              if((SkPi = 1) and (SkPj = 1)) then
                        count = count + 1
             if count >= minimum support count then
                  SSi.count ← count
            else
                  for m = i to max-1
                        SSm ← SSm+1
            max_temp ← max_temp-1
      BP ← SS
          max ← max_temp
} while max > 0
output BP
```

The pattern based collaborative filtering suggestion procedure generates the suggestion according to the student navigation pattern. The A priori algorithm is revised to discover the student navigation pattern and compute the Pearson correlation between each pattern and each student. The similarity formula is defined as equation 6:

$$corr(u,p) = \frac{\sum_{i=1}^{n}(MeanVi - \overline{MeanV}) \times (MeanPi - \overline{MeanP})}{n \sqrt{\frac{\sum_{i=1}^{n}(MeanVi - \overline{MeanV})^2}{n}} \times \sqrt{\frac{\sum_{i=1}^{n}(MeanPi - \overline{MeanP})^2}{n}}} \quad (6)$$

where *MeanVi* is the mean value of the student navigation vector, and *MeanPi* is the average mean value of the all user navigation vector with respect to the page *i*. $\overline{MeanV}$ is the absolute mean value of the pattern mean value vector, and $\overline{MeanP}$ is the absolute mean value of the pattern average vector.

If the correlation value is larger than zero, the page will be suggested to the students who have not read the one yet in the pattern. Therefore, the procedure is a pattern correlation suggestion approach and is different from the traditional collaborative filtering method.

Based on the proposed ideas above, the web page suggestion procedure and data flow is shown in Figure 4. The request and behavior information from students will be captured through the HTML Interface and recorded by the WPN, KFR Register. The information that contains the students' behavior will be normalized into the WKS Model. The information generated by the two procedures above will be stored into and retrieved from the Online Database. Then the two major procedures — Student Clustering and Navigation Pattern Discovery, will generate the student groups for each web page and discover the navigation pattern respectively. The results of the two procedures will be stored into and retrieved from the Suggestion Database.

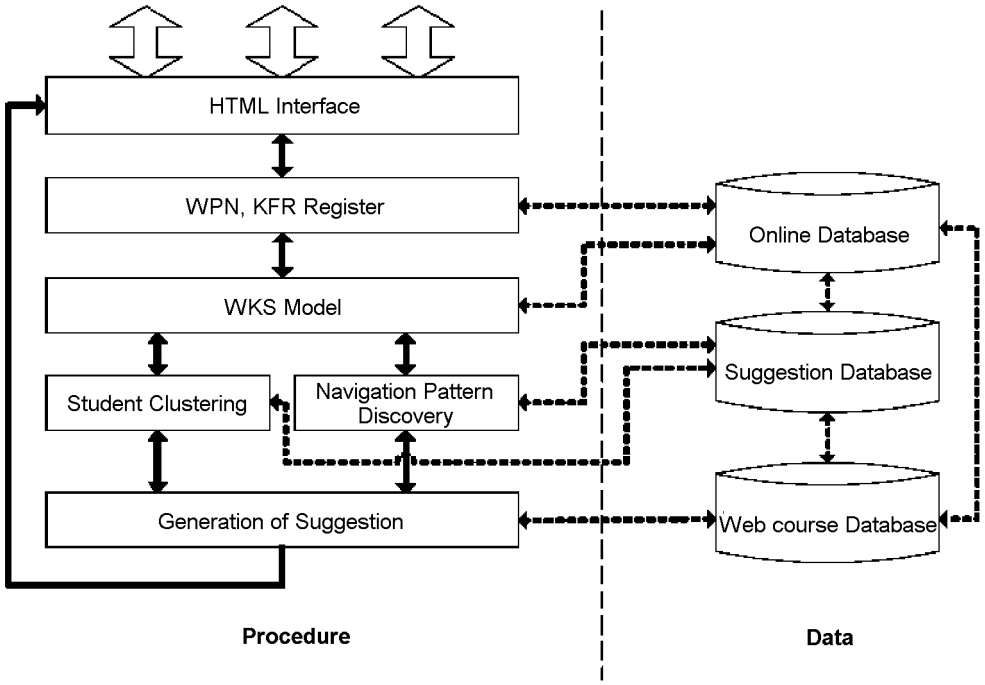

Figure 4. Web page navigation suggestion procedure and data flow.

6. Suggestion Trigger Events

In our system, we design several events that would trigger the suggestion procedure. The events and flowchart is shown in Figure 5.

Basically, the suggestion can be triggered by two sorts of events: one is for the entrance of the student, and the other is for the entrance of new web pages (content). We explain the events and flow as follows:

Case 1:

In this suggestion event, a member of the system log in the system, the suggestion would be constructed according to his KPN and KFR records and the WKS model. The student can choose to accept the suggestion or to reject it. If the student accepts the suggestion, he will get the suggested web page content; if the student rejects the suggestion, it indicates that the student might not be interested in the suggested information. This rejection would be feedback to the system for adjusting his parameters (vector) in the WKS model.

Case 2:

If a student who is new to the system, then we ask him to register as a member first. Because we don't have any of his navigation history, thus we don't have any information to refer to when we want to suggest web content to him. In this circumstance, the system would guide the student to have a personal interest poll — to select some keywords of interests prompted by the system. Through this process, the system can then build a polling vector similar to the navigation vector. Equation 7 is the calculation for the polling vector:

$$polling\_vector\_element_i = poll_i \times \frac{score_i}{\sum_{k=1}^{n} score_k} \tag{7}$$

where $poll_i$ indicates whether the student is interested in the keyword i or not (a Boolean value, "0" or "1"); $score_i$ indicates the weighted value of the keyword i which represents the degree of interest; n indicates the number of keywords in the system.

After having the polling of interest of a new student the suggestion process can then searching for suitable web page according to the similarity between he and other existing students — the new member's polling vector and the existing members' navigation vectors. Equation 8 is the calculation for the similarity measurement:

$$sim(n,u) = \frac{\vec{V}_n \cdot \vec{V}_u}{\left|\vec{V}_n\right|\left|\vec{V}_u\right|} \tag{8}$$

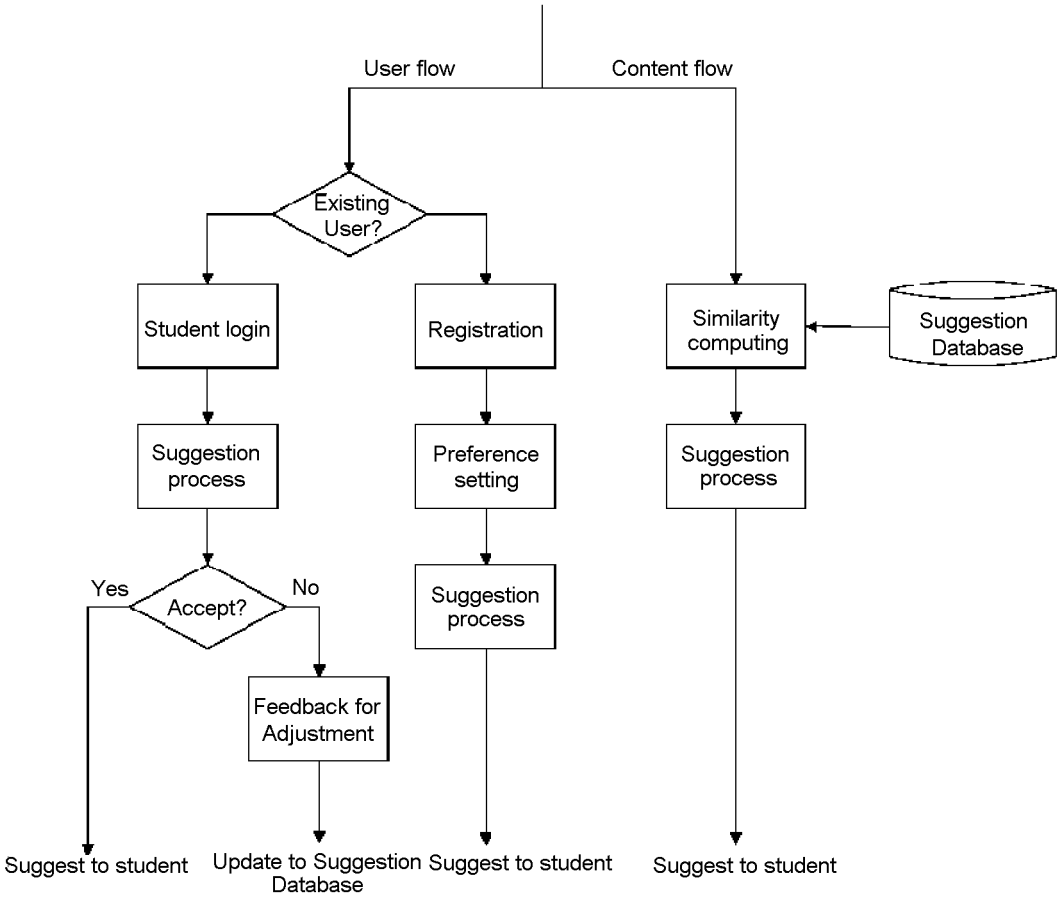

Figure 5. Suggestion events and flowchart.

where $\vec{V}_n$ is the polling vector of the new member; $\vec{V}_u$ is the navigation vector of an existing member.

Case 3:

Besides the circumstance of the entrance of a new member (user), there is another circumstance that a new web page (content) enters into the system. In this suggestion event, the similarities between the new coming web page and the existing pages are calculated. They can be obtained by the dis similarity matrix like the one shown in Table 5.

According to the Keyword-Webpage matrix, we have:

- *w:* the count of the common keyword(s) both page *i* and page *j* have;
- *x*: the count of the keyword(s) that page *j* has yet page *i* does not;
- *y:* the count of the keyword(s) that page *i* has yet page *j* does not;
- *z:* the count of the keyword(s) that neither page *i* nor page *j* has.

Table 5. A dissimilarity matrix

		Web page i		
		1	0	
Web page j	1	w	x	w+x
	0	y	z	y+z
		w+y	x+z	w+x+y+z

Then, we have the dissimilarity function calculated as equation 9:

$$dissim(i, j) = \frac{x + y}{w + x + y + z} \tag{9}$$

The larger value the *dissim(i,j)* produces, the less similarity page *i* and page *j* have; the smaller value the *dissim(i,j)* produces, the more similarity page *i* and page *j* have. The new coming web pages will be suggested to the students who have read the existing web pages that are similar to this new coming page. In other words, the

students who have read the existing pages that are similar to the new coming one are the candidate suggestion target of the new coming page.

7. Conclusion

Recommendation/suggestion Systems, allow users to share information about items they like or dislike and obtain suggestions based on predictions about unseen items in a timely fashion. In this process, users' preferences are considered to be the learning target functions.

A common problem of collaborative information recommender systems is that a collaborative filtering system requires a large group of users who have overlapping interests and that these users have interacted with the system for some time. The system will not be useful if it is sparsely used. Even if two users have agreed often on similar items, these two would not necessarily end up being nearest neighbors. This makes it difficult for the system to start making recommendations.

We alleviated this problem by integrating other techniques. In this research, we combined collaborative recommendation with techniques for recommending based on content analysis, to find content-based correlations between information items (web pages). The system generates a content representation for each of web page, and compute document similarity. The proposed Web Content Suggestion System for Distance Learning combines a content-based and a collaborative filtering content suggesting system filters web pages according to content analysis and creates usage profiles for student groups with similar interests.

The main idea of this research is through the utilization of information technology and the content of the web page resource, to not only provide what may be insufficient in the classroom teaching, but also through the automated web content suggestion procedure, to extend students' interests and enrich their knowledge. We collaborate with the lesson topics and web page resource to extend students' learning space. Instructor publishes supplementary reading or course relative extracurricular content to the learning system website. Students can browse and read the web pages according to their interests. And through our system, the students can share the interest and knowledge thus learning from each other in an implicit way.

References

[1] Cooley, R. and Srivastava, J., "Web Mining: Information and Pattern Discovery on the World Wide Web." *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97),* (1997).

[2] Yan, T. W. and Garcia-Molina, H., "The SIFT Information Dissemination System." *ACM Transactions on Database Systems (TODS),* Vol. 24, pp. 529– 565.

[3] Mostafa, J. M., Mukhopadhyay, S., Lam, W. and Palakal, M., "A Multilevel Approach to Intelligent Information Filtering: Model, System and Evaluation." *ACM Transaction of Information System,* Vol. 15 (1997).

[4] Melnik, S., Raghavan, S., Yang, B. and Garcia-Molina, H., "Building a Distributed Full-Text Index for the Web*." ACM Transactions on Information Systems (TOIS),* Vol. 19, pp. 217–241, (2001).

[5] Srivastava, J., Cooley, R., Deshpande, M. and Tan, P., "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data." *SIGKDD Explorations,* Vol. 1, pp. 12–23, (2000).

[6] Tan, K. W., Han, H. and Elmasri, R., "Web Data Cleansing and Preparation for Ontology Extraction Using WordNet." *First International Conference on Web Information Systems Engineering (WISE'00),* Vol. 2.

[7] Kosala, R. and Blockeel, H., "Web Mining Research: A Survey." *ACM SIGKDD,* Vol. 2, pp. 1–15, (2000).

[8] Ester, M., Kriegel, H. P., Sander, J., Wimmer, M. and, Xu, X., "Incremental Clustering for Mining in a Data Warehousing Environment." *Proceeding of 24th Int. Conf. Very Large Databases (VLDB'98),* New York, NY, USA. pp. 24–27 (1998).

[9] Guinepain, Sylvain and Gruenwald, Le., "Research Issues in Automatic Database Clustering." *ACM SIGMOD Record,* Vol. 34 (2005).

[10] Perkowitz, M. and Etzioni, O., "Adaptive Sites: Automatically Learning from User Access Patterns." *Proceeding of the Sixth International WWW Conference,* Santa Clara, CA. (1997).

[11] Mobasher, B., Cooley, R. and Srivastava, J., "Creating Adaptive Web Sites through Usage-Based Clustering of Urls." *IEEE Knowledge and Data Engineering Workshop (KDEX'99)* (1999).

[12] Mobasher, B., Dai, H., Luo, T. and Nakagawa, M., "Effective Personalization Based on Association Rule Dis-

covery from Web Usage Data." *Proceeding of the 3rd ACM Workshop on Web Information and Data Management (WIDM01),* Atlanta (2001).

[13] Masseglia, F., Poncelet P. and Teisseire, M., "Using Data Mining Techniques on Web Access Logs to Dynamically Improve Hypertext Structure." *ACM SigWeb Letters,* Vol. 8, pp. 13–19, (1999).

[14] Kitsuregawa, M., Shintani, T. and Pramudiono, I., "Web Mining and Its SQL Based Parallel Execution." *IEEE Workshop on Information Technology for Virtual Enterprises* (2001).

[15] Srikant, R. and Yang, Y. "Mining Web Logs to Improve Website Organization." *Proceeding of the tenth International World Wide Web Conference,* Hong Kong (2001).

[16] Hirsh, H., Basu, C. and Davison, B. D., "Learning to Personalize." *Communications of the ACM,* Vol. 43, pp. 102–106 (2000).

[17] Smith, B. and Cotter, P., "A Personalized Television Listings Service." *Communications of the ACM,* Vol. 43, pp. 107–111 (2000).

[18] Joachims, T., Freitag, D. and Mitchell, T., "Web Watcher: A Tour Guide for the World Wide Web." *Proceedings of Joint Conference on Artificial Intelligence* (1997).

[19] Kantor, P. B., Boros, E. et al., "Capturing Human Intelligence in the Net." *Communications of the ACM,* Vol. 43, pp. 112–115 (2000).

[20] Chen, M. S., Han, J. W. and Yu, P. S., "Data Mining: An Overview from Database Perspective." *IEEE Transactions on Knowledge and Data Engineering,* Vol. 5, pp. 926–938 (1996).

[21] Perkowitx, M. and Etzioni, O., "Adaptive Web Sites." *Communications of the ACM,* Vol. 43, pp. 152–158 (2000).

[22] Spiliopoulou, M. "Web Usage Mining for Web Site Evaluation." *Communications of the ACM,* Vol. 43, pp. 127–134 (2000).

[23] Fu, Y., Creado, M. and Ju, C., "Reorganizing Websites Based on User Access Patterns." *Proceeding of the ACM CIKM International Conference on Information and Knowledge Management,* Atlanta, Georgia, USA, pp. 583–585 (2001).

Manuscript Received: Jul. 20, 2005
Accepted: Nov. 2, 2005